AMENDMENTS TO THE CLAIMS

Claims 1-3, 5-18, 20, 63-73 and 75-89 are pending. Please amend claims 1, 5, 12, 71, 75 and 82 as follows, without acquiescence or prejudice to pursue the original claims in a related application. Please cancel claims 4 and 74 as follows, without acquiescence or prejudice to pursue in a related application. A complete listing of the current pending claims is provided below and supersedes all previous claims listing(s).

1. (Currently Amended) A method of circuit equivalence checking, comprising:

solving at least a first equivalence checking problem;

storing at least a first solution to at least the first equivalence checking problem; and

reusing at least part of the first solution to the first equivalence checking problem for a second equivalence checking problem based upon information related to the first solution,

wherein the information related to the first solution comprises ~~includes~~ one or more algorithm traces or one or more problem signatures.

2. (Original) The method of claim 1, wherein at least one of the first equivalence checking problem and the second equivalence checking problem require determination of logical equivalence between at least combinational circuits.

3. (Original) The method of claim 1, wherein at least one of the first equivalence checking problem and the second equivalence checking problem require determination of logical equivalence between at least sequential logic circuits.

4. (Cancelled).

5. (Currently Amended) The method of claim 1 [[4]], wherein the one or more problem signatures include pre-cached problem signatures.

6. (Original) The method of claim 5, wherein the pre-cached problem signatures includes one or more numbers of inputs.

7. (Original) The method of claim 5, wherein the pre-cached problem signatures includes one or more numbers of intermediate signals.

8. (Original) The method of claim 5, wherein the pre-cached problem signatures includes one or more counts of logic gates.

9. (Original) The method of claim 8, wherein the pre-cached problem signatures includes one or more counts of logic gates of different types.

10. (Original) The method of claim 5, wherein the pre-cached problem signatures includes RTL information.

11. (Original) The method of claim 5, wherein the pre-cached problem signatures includes hierarchical information.

12. (Currently Amended) The method of claim 1 [[4]], wherein the one or more problem signatures include cached problem signatures.

13. (Original) The method of claim 12, wherein the cached problem signatures includes one or more numbers of inputs.

14. (Original) The method of claim 12, wherein the cached problem signatures includes one or more numbers of intermediate signals.

15. (Original) The method of claim 12, wherein the cached problem signatures includes one or more counts of logic gates.

16. (Original) The method of claim 15, wherein the cached problem signatures includes one or more counts of logic gates of different types.

17. (Original) The method of claim 12, wherein the cached problem signatures includes RTL information.

18. (Original) The method of claim 12, wherein the cached problem signatures includes hierarchical information.

19. (Cancelled).

20. (Original) The method of claim 1, wherein the first solution includes one or more stored problems.

21-62. (Cancelled).

63. (Original) An apparatus for circuit equivalence checking, comprising:

one or more persistent caches including:

one or more cached objects including:

one or more problem signatures; and

one or more algorithm traces;

and one or more cache managers accepting one or more equivalence checking queries and returning at least part of at least one cached object from the one or more persistent caches at least partly in response to the one or more equivalence checking queries.

64. (Previously Presented) The apparatus of claim 63, wherein the one or more persistent caches is populated with at least one cached object prior to the one or more cache managers accepting one or more equivalence checking queries.

65. (Previously Presented) The apparatus of claim 63, wherein at least partly in response to the one or more equivalence checking queries, the one or more persistent caches is updated.

66. (Previously Presented) The apparatus of claim 65, wherein the one or more persistent caches is updated at least by adding at least one cached object to the one or more persistent caches.

67. (Previously Presented) The apparatus of claim 65, wherein the one or more persistent caches is updated at least by removing at least one cached object from the one or more persistent caches.

68. (Previously Presented) The apparatus of claim 65, wherein the one or more persistent caches is updated at least by changing at least one cached object in the one or more persistent caches.

69. (Previously Presented) The apparatus of claim 65, wherein the one or more persistent caches is updated at least by replacing at least one cached object in the one or more persistent caches.

70. (Previously Presented) The apparatus of claim 63, wherein the one or more persistent caches are searched with low cost algorithms prior to high cost algorithms.

71. (Currently Amended) A computer readable medium comprising a computer program stored in a storage medium or a memory medium which, when executed by a processing system, causes the system to perform a method of circuit equivalence checking, the method comprising:

solving at least a first equivalence checking problem;

storing at least a first solution to at least the first equivalence checking problem; and

reusing at least part of the first solution to the first equivalence checking problem for a second equivalence checking problem based upon information related to the first solution,

wherein the information related to the first solution comprises ~~includes~~ one or more algorithm traces or one or more problem signatures.

72. (Previously Presented) The medium of claim 71, wherein at least one of the first equivalence checking problem and the second equivalence checking problem require determination of logical equivalence between at least combinational circuits.

73. (Previously Presented) The medium of claim 71, wherein at least one of the first equivalence checking problem and the second equivalence checking problem require determination of logical equivalence between at least sequential logic circuits.

74. (Cancelled).

75. (Currently Amended) The medium of claim 71 [[74]], wherein the one or more problem signatures include pre-cached problem signatures.

76. (Previously Presented) The medium of claim 75, wherein the pre-cached problem signatures includes one or more numbers of inputs.

77. (Previously Presented) The medium of claim 75, wherein the pre-cached problem signatures includes one or more numbers of intermediate signals.

78. (Previously Presented) The medium of claim 75, wherein the pre-cached problem signatures includes one or more counts of logic gates.

79. (Previously Presented) The medium of claim 78, wherein the pre-cached problem signatures includes one or more counts of logic gates of different types.

80. (Previously Presented) The medium of claim 75, wherein the pre-cached problem signatures includes RTL information.

81. (Previously Presented) The medium of claim 75, wherein the pre-cached problem signatures includes hierarchical information.

82. (Currently Amended) The medium of claim 71 [[74]], wherein the one or more problem signatures include cached problem signatures.

83. (Previously Presented) The medium of claim 82, wherein the cached problem signatures includes one or more numbers of inputs.

84. (Previously Presented) The medium of claim 82, wherein the cached problem signatures includes one or more numbers of intermediate signals.

85. (Previously Presented) The medium of claim 82, wherein the cached problem signatures includes one or more counts of logic gates.

86. (Previously Presented) The medium of claim 85, wherein the cached problem signatures includes one or more counts of logic gates of different types.

87. (Previously Presented) The medium of claim 82, wherein the cached problem signatures includes RTL information.

88. (Previously Presented) The medium of claim 82, wherein the cached problem signatures includes hierarchical information.

89. (Previously Presented) The medium of claim 71, wherein the first solution includes one or more stored problems.